Computer Science 385
Design and Analysis of Algorithms
Siena University
Spring 2026

Heaps Practice

Recall some definitions:

- A *full* binary tree of height h:
- A *complete* binary tree of height h
- We say T is *balanced* if
- **Lemma**: If T is a binary tree, then at level k, T has ________ nodes.
- **Theorem**: If T has height h then n = num nodes in T __________
  Equivalently, if T has n nodes, then

We will look at an example of a linked representation and an array representation of an example tree.

The children of node i are stored in positions ______ and ______, and therefore the parent of a node j, may be found at ______.

Priority Queues

Efficiency of priority queue operations when implementations are based on traditional list structures (array/`ArrayList` or singly-linked list):

We can do better with a structure called a _______.

Definition: A *Min-Heap* H is

Or, thinking in terms of array indices:

Example min-heap

This is the perfect building block for a priority queue!

Inserting into a Heap

1. Place number to be inserted at the next free position.
2. “Percolate” it up to correct position.

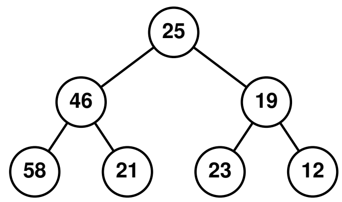
Deleting the Root from a Heap

1. Save value in root element for return (it's the smallest one).
2. Move last element to root
3. Push down (or "sift") the element now in the root position (it was formerly the last element) to its correct position by repeatedly swapping it with the smaller of its two children.

Efficiencies of heap `add` and `remove`:

Sorting with a Heap (HeapSort)

"Heapify" steps:



Heapify total efficiency: $\Theta(\qquad)$.

See the notes for a detailed analysis to justify this.

Completing the Heapsort

Efficiency of the entire process of extracting elements in sorted order is $\Theta(\qquad)$.

Therefore, the total time is $\Theta(\qquad)$.

Space overhead: